Docket No.: RSW920000039US1 (7161-096U) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Customer Number: 46320
Christopher GAGE, et al.	:	Confirmation Number: 8638
Application No.: 09/557,708	:	Group Art Unit: 2141
Filed: April 25, 2000	:	Examiner: K. Shingles

For: URL BASED STICKY ROUTING TOKENS USING A SERVER SIDE COOKIE JAR

SUPPLEMENTAL APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Supplemental Appeal Brief is submitted, pursuant to 37 C.F.R. § 41.39(b)(2), in support of the Appeal Brief filed September 19, 2005, in response to the Examiner reopening prosecution in the Office Action dated April 12, 2006, and in further response to the Examiner reopening prosecution in the Office Action dated December 19, 2006, wherein Appellants appeal from the Examiner's rejection of claims 1-27.

I. REAL PARTY IN INTEREST

This application is assigned to IBM Corporation by assignment recorded on August 22, 2000, at Reel 011152, Frame 0250.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1-27 are pending and finally rejected in this Application. It is from the final rejection of claims 1-27 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

The claims have not been amended subsequent to the imposition of the Final Office Action dated March 18, 2005 (hereinafter the Third Office Action), or the reopening of prosecution by the Examiner in the Office Action dated April 12, 2006 (hereinafter the Fourth Office Action), or the reopening of prosecution by the Examiner in the Office Action dated December 19, 2006 (hereinafter the Fifth Office Action). Although a Response was submitted with respect to the Third Office Action pursuant to the provisions of 37 C.F.R. § 1.116 on May 19, 2005, this Response did not make any changes or additions to the claims.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Referring to claims 1 and 12 and Figures 1 and 4A, 4B of Appellants' specification, a method of establishing a persistent relationship between an end user device 101, 103 and a server 109 where the server 109 is one of a plurality of servers 109 managed by a dispatcher 107 and the end user device 101, 103 accesses the server 109 using a uniform resource locator (URL) is disclosed. In step 401, a request for information from the end user device is received at the dispatcher 107, and the dispatcher 107 determines which of a plurality of servers 109 to select for satisfying the request (page 10, lines 12-15). In step 403, a token 235 is created at the selected server 109, and the token 235 includes at least an identifier 207 for the selected server 109, a date/time stamp 209, and a key 211. The key 211 accesses a server-side storage area for information regarding the persistent relationship and the end user device (page 10, lines 16-25).

In step 437, the token 235 is inserted into the URL (page 12, lines 1-9). In step 439, a response, with the token 235 inserted into the URL, is sent by the selected server 109 to the client device 101, 103.

Referring to independent claims 7 and 18 and Figures 1 and 3 of Appellants' specification, a method of routing a request by an end user device 101, 103 to a particular one of a plurality of redundant servers 109 residing behind a network dispatching mechanism 107 is disclosed. In step 301, a request for information indicated by a uniform resource locator (URL) is received at the network dispatching mechanism 107 (page 12, line 16). In step 303, the network dispatching mechanism 107 determines if the URL contains a valid routing token 235 (page 12, lines 17-18). In step 311, a determination is made at the network dispatching mechanism 107 as to whether the session binding indicated by the routing token 235 is old (page 12, lines 21-22). In step 313, if the routing token 235 is not old, the network dispatching mechanism 107, forwards the request, including the URL, to the particular server 109 indicated by the valid routing token 235 (page 12, lines 26-27).

The valid routing information from the URL is removed by the particular server 109 (page 13, line 6). The particular server 109 stores the routing information removed from the valid routing token 235, and the valid routing information can be accessed subsequently by an outbound data stream filter during the processing of an outbound reply related to the request (page 13, lines 6-7). The particular server 109 accesses a server-side storage location where session information regarding a session between the particular server 109 and the end user device 101, 103 is stored, and the accessed session information is inserted into the request (page 9, lines 11-5).

Referring to independent claims 10 and 20 and Figure 4B, a method of sending

information to a requesting end user 101, 103 from an application over a session wherein the application resides at one of a plurality of redundant servers 109 residing behind a network dispatcher 107 is disclosed. In step 421, response information including a URL (uniform resource locator) is received from the application (page 11, lines 8-10). In step 423, a determination is made if a server-side key cookie has been used for storing session information between the end user 101, 103 and the application (page 11, lines 10-14). In step 425, if server-side key cookie has been used for storing session information, a session key 211 from the key cookie is retrieved (page 11, line 14). In step 426, if a key cookie was not used for storing session information, a session key from a control block is retrieved. In step 427, all cookies are removed from the response information (page 10, lines 14-15). In step 429, the removed cookies are stored in a predetermined server-side storage area (page 11, lines 14-16).

In step 431, the URL is updated to indicate the removal of the cookies (page 11, lines 20-27). In step 433, a sticky routing string is created (page 11, line 27 through page 12, line 1). In step 435, a date/time stamp in the sticky routing string is updated page 11, line 27 through page 12, line 1). In step 437, the sticky routing string is inserted into the URL (page 12, lines 1-8). In step 439, the response information, including the URL, is transmitted to the end user 101.103 (page 12, lines 8-9).

Referring to independent claim 22 and Figures 1-2 and 4A, 4B of Appellants' specification, a network dispatcher 107 for establishing a persistent relationship between an end user device 101, 103 and a server 109 where the server 109 is one of a plurality of servers 109 managed by the network dispatcher 107 is disclosed. Means are included for receiving a request for information from the end user device at the dispatcher, and means are included to determine which of a plurality of servers 109 to select for satisfying the request (page 10, lines 12-15).

Means are included for creating the token 235, which includes at least an identifier 207 for the selected server 109, a date/time stamp 209, and a key 211. The key 211 accesses a server-side storage area for information regarding the persistent relationship and the end user device 101, 103 (page 10, lines 16-25). Means are included for inserting the token 235 into the URL (page 12, lines 1-9), and means are included for sending, by the selected server 109, a response, with the token 235 inserted into the URL, to the client device 101, 103.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 12, and 22 were rejected under the second paragraph of 35 U.S.C. § 112;

2. Claims 1-5, 9, 12-16, and 22-26 were rejected under 35 U.S.C. § 103 for obviousness based upon Schmeidler et al., U.S. Patent No. 6,763,370 (hereinafter Schmeidler), in view of Kunzelman et al., U.S. Patent No. 6,041,357 (hereinafter Kunzelman);

3. Claims 7-8 and 18-19 were rejected under 35 U.S.C. § 103 for obviousness based upon Schmeidler in view of Kunzelman and further in view of Lee et al., U.S. Patent No. 6,609,150 (hereinafter Lee); and

4. Claims 6, 10-11, 17, 20-21, and 27 rejected under 35 U.S.C. § 103 for obviousness based upon Gupta et al., U.S. Patent No. 6,763,468 in view of Schmeidler and Kunzelman.

VII. ARGUMENT

THE REJECTION OF CLAIMS 1, 12, AND 22 UNDER THE SECOND PARAGRAPH OF 35 U.S.C. § 112

For convenience of the Honorable Board in addressing the rejections, claims 12 and 22 stand or fall together with independent claim 1.

On page 3 of both the Fourth Office Action and the Fifth Office Action, the Examiner asserted the following:

> Claims 1, 12 and 22 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential structural cooperative relationships of elements, such omission amounting to a gap between the necessary structural connections. See MPEP § 2172.01.

The Examiner then proceeded to list four "structural cooperative relationships preceding the insertion of the token into the URL" that the Examiner asserted were omitted.

For ease of reference, M.P.E.P. § 2172.01 is reproduced below:

> A claim which omits matter disclosed to be essential to the invention as described in the specification or in other statements of record may be rejected under 35 U.S.C. 112, first paragraph, as not enabling. *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976). See also MPEP § 2164.08(c). Such essential matter may include missing elements, steps or necessary structural cooperative relationships of elements described by the applicant(s) as necessary to practice the invention.
>
> In addition, a claim which fails to interrelate essential elements of the invention as defined by applicant(s) in the specification may be rejected under 35 U.S.C. 112, second paragraph, for failure to point out and distinctly claim the invention. See *In re Venezia*, 530 F.2d 956, 189 USPQ 149 (CCPA 1976); *In re Collier*, 397 F.2d 1003, 158 USPQ 266 (CCPA 1968). >But see *Ex parte Nolden*, 149 USPQ 378, 380 (Bd. Pat. App. 1965) ("[I]t is not essential to a patentable combination that there be interdependency between the elements of the claimed device or that all the elements operate concurrently toward the desired result"); *Ex parte Huber*, 148 USPQ 447, 448-49 (Bd. Pat. App. 1965) (A claim does not necessarily fail to comply with 35 U.S.C. 112, second paragraph where the various elements do not function simultaneously, are not directly functionally related, do not directly intercooperate, and/or serve independent purposes.).<

At the outset, Appellants note that the case law referred to above generally applies to "kit" or "assembly" claims. Independent claims 1, 12, and 22, however, are respectively directed to a method, a computer program product, and a network dispatcher, which on their face do not appear to be kit or assembly claims.

Moreover, as summarized in above-reproduced passages, the case law requires that the alleged omitted essential matter be identified by Appellants as "essential" within the "specification" or within "other statements of record." The Examiner, however, has failed to establish that the four features alleged by the Examiner to be omitted, have been identified, by

Appellants, as being essential.

Furthermore, Appellants surmise that the Examiner has misinterpreted the enablement requirement of the first paragraph of 35 U.S.C. § 112, as the Examiner appears to be requiring that the independent claims enable the invention. This requirement, however, is not consonant with the first paragraph of 35 U.S.C. § 112, which only requires that the specification describe how to make and use the invention.

The above arguments were previously presented on pages 6 and 7 of the Second Appeal Brief filed September 21, 2006. However, in reopening prosecution in the Fifth Office Action, the Examiner did not respond to these arguments despite repeating the Examiner's prior rejection of claims 1, 12, and 22 under the second paragraph of 35 U.S.C. § 112.

Appellants respectfully submit that the Examiner has failed to establish a proper rejection under the second paragraph of 35 U.S.C. § 112 for the reasons set forth above and in the Second Appeal Brief.

THE REJECTION OF CLAIMS 1-5, 9, 12-16, 22-26 UNDER 35 U.S.C. § 103 FOR OBVIOUSNESS BASED UPON SCHMEIDLER IN VIEW OF KUNZELMAN

For convenience of the Honorable Board in addressing the rejections, claims 2-5, 9, 12-16, and 22-26 stand or fall together with independent claim 1.

At the outset, Appellants note that the Examiner has failed to clearly designate the teachings in Schmeidler being relied upon the statement of the rejection. In this regard, the Examiner's rejection under 35 U.S.C. § 103 also fails to comply with 37 C.F.R. § 1.104(c), which reads:

> In rejecting claims for want of novelty or for obviousness, the examiner must cite the best references at his or her command. When a reference is complex or shows or describes inventions other than that claimed by the applicant, the particular part relied on must be designated as nearly as practicable. The pertinence of each reference, if not apparent, must be clearly explained and each rejected claim specified.

Referring to independent claim 1: (i) a dispatcher receives a request from a client device, (ii) the dispatcher selects a server for satisfying the request, (iii) a token is created at the selected server, and (iv) the token is sent by the selected server to the client device. As best can be understood from the Examiner's rejection, the SCDP client 216 of Schmeidler corresponds to the claimed client device (see the Examiner's cited passages of column 9, lines 5-8; page 13, lines 42-53; and page 23, lines 13-15, which all refer to SCDP client).

On page 4 of the Fifth Office Action, the Examiner asserted the following:

> determining by the dispatcher, which of the plurality of server to select for satisfying the request (col.24 lines 15-22 and 31-43 — CAS server determines the appropriate RAFT server for satisfying the request).

Based upon this statement, Appellants proceed on the basis that the Examiner is alleging that CAS server 210 corresponds to the claimed dispatcher and that RAFT server 206 corresponds to the claimed selected server for satisfying the response.

Based upon these assumptions, Appellants note that the Examiner has made assertions regarding the teachings of Schmeidler that are factually incorrect. The CAS server does not determine the appropriate RAFT server 206 for satisfying the request. In this regard, reference is

made to the Examiner's cited passage of column 24, lines 31-43:

> Launcher then examines the list of URLs to determine if any RAFT URLs are present. If RAFT URLs are present, the Launcher sends only the list of RAFT URLs along with the RAFT access token to ARFSD VxD which will forward this information to the RAFT client, i.e. the RAFT VxD of the SCDP client. The Launcher also provides a weight for each of the RAFT URLs. These weights may be different than the ones provided by the CAS during the URN to URL conversion. The RAFT client then establishes a connection with one of the RAFT servers specified by the list of URLs.
>
> The RAFT client may contain the appropriate program logic which enables it to use the weights provided with the URLs to decide which RAFT server to contact first.

As described above the "Launcher" receives the list of URLs and then based upon some unknown determination, "[t]he RAFT client then establishes a connection with one of the RAFT servers specified by the list of URLs." Neither the Launcher nor the RAFT client are associated with the CAS server 210 (i.e., the alleged dispatcher). Instead, as illustrated in Figs. 2A and 3A and described in column 7, lines 15-31, both the RAFT Client VxD 222 and the Launcher 220 are found within the SCDP client 216 (i.e., the alleged client). Thus, the functions that the Examiner alleges are performed by the CAS server 210 (i.e., the alleged dispatcher) are actually performed by the SCDP client 216 (i.e., the alleged client). Thus, Schmeidler fails to teach the above-identified limitation for which Schmeidler is being relied upon to teach.

Appellants also note that another inconsistency exists between what the Examiner alleges Schmeidler to teach and what Schmeidler actually teaches. On page 4 of the Fifth Office Action, the Examiner further asserted the following:

> creating, at the selected server, a token comprising at least an identifier for the selected server, a date/time stamp, and a key, said key for accessing a server-side storage area for information regarding the persistent relationship at the end user device (Figures 8, 9 and 13; col.9 lines 50-5 1, col.14 lines 43-45, col. 15 lines 23-32, col.18 lines 37-42, col.19 lines 21-23, col.22 lines 41-66, col.30 lines 21-41—-the activator functions as the key and includes a RAFT token for accessing a particular RAFT server, wherein the token also comprises the URN which is an identifier for the selected server and a timestamp/expiration time).

Appellants are entirely unclear from the statement of the rejection as to what feature within Schmeidler is being relied upon by the Examiner to identically disclose the claimed server-side

storage area. Notwithstanding this failure by the Examiner to clearly identify the claimed elements within the applied prior art being relied upon in the statement of the rejection, the Examiner's cited passage teaches away from the claimed invention.

The first limitation found within this clause is that the token is created at the selected server, and as already noted above, the Examiner is relying upon the RAFT server 206 to teach the claimed selected server. However, Schmeidler clearly and unambiguously teaches that the CAS 210 generates both the activator and the RAFT token, which the Examiner is relying upon to teach the claimed token. For example, the Examiner's cited passage of column 14, lines 42-45 states the following:

> In response to the request for purchase authorization, CAS 210 generates an activator, including a RAFT token, which is transmitted through the secure RPC connection to the SCDP client 216.

Thus, the CAS server 210 (i.e., the alleged dispatcher) generates both the activator and RAFT token. Reference is also made to the Examiner's cited passages of column 15, lines 23-32 and column 22, lines 41-47:

> The RAFT token received from the CAS 210 includes an end time field as described with reference to FIG. 8 and its accompanying description. Prior to expiration of the activator and RAFT token, the launcher module 220 issues a request via a secure RPC connection to CAS server 206 for a refreshed activator/RAFT token pair, as illustrated by decisional step 534 and process step 536. The new activator/RAFT token pair are installed and utilized in a manner similar to that previously described, as illustrated by process step 538. (column 15, lines 23-32)
>
> To improve the overall security model of the SCDP system, the CAS provides the SDCP client with a signed RAFT Authorization Token. The RAFT token authorizes a particular SCDP client to access a particular URN, for a specified time period. The CAS digitally signs the RAFT Token, using standardized, public-key digital signature algorithms. (column 22, lines 41-47)

As noted above, claim 1 recites that the token is created at the selected server. However, as clearly described above, neither the activator nor the RAFT token are created at the alleged selected server (i.e., RAFT server 206). Instead, the activator and the RAFT token are created at the alleged dispatcher (i.e., CAS server 210). Thus, not only does Schmeidler fail to teach the above-identified additional limitation for which Schmeidler is being relied upon to teach,

Schmeidler teaches away from the claimed invention.

Therefore, for the reasons stated above, even if one having ordinary skill in the art were motivated to modify Schmeidler in view of Kunzelman, the claimed invention would not result.

THE REJECTION OF CLAIMS 7-8 AND 18-19 UNDER 35 U.S.C. § 103 FOR OBVIOUSNESS BASED UPON SCHMEIDLER IN VIEW OF KUNZELMAN AND LEE

For convenience of the Honorable Board in addressing the rejections, claims 8 and 18-19 stand or fall together with independent claim 7.

At the outset, Appellants note that the Examiner's rejection of claims 7-8 and 18-19 has also failed to clearly designate the teachings in Schmeidler being relied upon the statement of the rejection, as required by 37 C.F.R. § 1.104(c).

Independent claim 7 recites the following:

> receiving, at the network dispatching mechanism, a request for information indicated by a uniform resource locator (URL);
>
> determining, at the network dispatching mechanism, if said URL contains a valid routing token.

On page 7 of the Fifth Office Action, with regard to the network dispatch mechanism receiving the request, the Examiner cited the same passages that the Examiner cited to teach the claimed dispatcher receiving a request, as recited in claim 1. Thus, Appellants proceed on the basis that the Examiner intended to rely upon the same feature (i.e., CAS server 210) in Schmeidler to teach both the network dispatching mechanism, as recited in claim 7, and the dispatcher, as

recited in claim 1.

With regard to the second limitation reproduced above, the Examiner stated the following on page 7 of the Fifth Office Action:

> determining, at the network dispatching mechanism, if said URL contains a valid routing token (col.23 lines 1-5, col.24 lines 44-67).

For ease of reference, the Examiner's cited passage of column 23, lines 1-5 of Schmeidler is reproduced below:

> The CAS signs the token with the CAS group's private key so that the RAFT server can validate its authenticity. The RAFT server will deny access if server's current time is not within the token's window.

As evident from this passage, the CAS server 210 (i.e., the alleged network dispatch mechanism) does not determine if the URL contains a valid routing token. Instead, the RAFT server validates the authenticity of the token. As noted with regard to claim 1, the Examiner is relying upon the RAFT server 206 to teach the claimed selected server, and the Examiner is presumably alleging that the RAFT server 206 also teaches the claimed "particular server indicated by said valid routing token" recited in claim 7. Thus, Schmeidler fails to teach the above-identified limitation for which Schmeidler is being relied upon to teach.

Appellants also note that several other inconsistencies exist between what the Examiner alleges Schmeidler to teach and what Schmeidler actually teaches. In this regard reference is made to the following assertion in the third bullet on page 7 of the Fifth Office Action:

> if said URL contains a valid routing token, further determining, at the network dispatching mechanism, if a session binding indicated by said routing token is old (col.25 lines 1-20, col.26 lines 4-13)

For ease of reference, the Examiner's cited passage of column 25, lines 1-10 of Schmeidler is reproduced below:

> Eventually, the RAFT access token will expire. The SCDP client's activator keep-alive mechanism is responsible for obtaining a new RAFT token before the current token expires. This insures RAFT tokens are refreshed in timely manner so that access failures will not occur under normal operating conditions. When the RAFT client sends the token to the RAFT server during a RAFT_OPEN, RAFT client must compute how long the token is valid from the start and expiration time. Since the RAFT client cannot verify the legitimacy of the token contents without the CAS' public key, RAFT client must wait for a successful RAFT_OPEN to determine that the token is valid before setting its refresh time. However, the refresh time is based upon when RAFT client received the token and not when the RAFT_OPEN completed. To insure uninterrupted access to the server, the RAFT client requests a new RAFT access token from the CAS in advance of when the token expires. Upon receipt of the new RAFT access token, the RAFT client will send a RAFT_REFRESH operation with the newly obtained token to the RAFT server.

As evident from this passage, the CAS server 210 (i.e., the alleged network dispatch mechanism) does not determine if a session binding indicated by the routing token is old. This passage is complete silent with regard to the alleged network dispatch mechanism (i.e., CAS server 210). Instead, this passage within Schmeidler teaches that the refreshing of the RAFT access token is accomplished with the SCDP client (i.e., the alleged client). Thus, Schmeidler fails to teach the above-identified additional limitation for which Schmeidler is being relied upon to teach.

Reference is made to the following assertion in the fourth bullet on page 7 of the Fifth Office Action:

> if said URL contains a valid routing token and said routing token is not old, forwarding, by said network dispatching mechanism, the request, including the URL, to the particular server indicated by said valid routing token (col. 24 lines 41-65).

For ease of reference, the Examiner's cited passage of column 24, lines 41-65 of Schmeidler is reproduced below:

> The RAFT client may contain the appropriate program logic which enables it to use the weights provided with the URLs to decide which RAFT server to contact first.
>
> The RAFT client then attempts to open a Briq on the RAFT server 1000. The client specifies a protocol version, the path name (from the URL) and the RAFT access token. The protocol version is a 32-bit value used to verify that the RAFT client and RART server are protocol compatible. To validate access, the RAFT server verifies that the URN provided in the token is one of the ones listed in the Briq header. The RAFT server 1000 checks the RAFT token's start and expiration times during the open. If the RAFT_OPEN is successful, the RAFT server

> returns a RAFT file handle and a unique ID for the Briq, e.g. a hash of the Briq tag, used for caching.
>
> In order for the RAFT server to validate the expiration time, the RAFT server time is synchronized with the CAS to within a predetermined interval. The RAFT server therefore accepts start times earlier than the current time and does not deny access until after expiration of the interval. The token expiration time is proposed to be some multiple of the Activator keep-alive time plus additional time to handle varying network and server latencies.
>
> On each subsequent RAFT read request, the RAFT server checks that the access token has not expired. The RAFT server will fail any request that occurs when the server does not have a valid access token for that particular client.
>
> Eventually, the RAFT access token will expire. The SCDP client's activator keep-alive mechanism is responsible for obtaining a new RAFT token before the current token expires.

As evident from this passage, the CAS server 210 (i.e., the alleged network dispatch mechanism) does not forward the request, including the URL, to the particular server indicated by the routing token. This passage is complete silent with regard to the alleged network dispatch mechanism (i.e., CAS server 210). Thus, Schmeidler fails to teach the above-identified additional limitation for which Schmeidler is being relied upon to teach.

Reference is made to the following assertion in the fifth bullet on page 7 of the Fifth Office Action:

> removing, by said particular server, said valid routing information from the URL (col.13 lines 50-54).

For ease of reference, the Examiner's cited passage of column 13, lines 49-54 of Schmeidler is reproduced below:

> Upon invocation, Launcher 220 extracts the Universal Resource Name (URN) from the Launch String and requests the CAS 210 to perform a URN to URL conversion, as illustrated by step 6. The URN of the present invention is a unique identifier of a title within a briq.

As evident from this passage, the Launcher 220 extracts the URN from the Launch String, which presumably allegedly corresponds to the claimed "removing, by said particular server, said valid routing information from the URL." However, as previously noted, the Launcher 220 is part of the SCDP client 216 (i.e., the alleged client) and not part of the alleged particular server (presumably RAFT server 206). Thus, Schmeidler fails to teach the above-identified additional

limitation for which Schmeidler is being relied upon to teach.

Reference is made to the following assertion in the sixth bullet on page 7 of the Fifth Office Action:

> storing, by said particular server, said routing information removed from said valid routing token, where said valid routing information can be accessed subsequently by an outbound data stream filter during the processing of an outbound reply related to said request (col.13 line 50-col.14 line 22).

Upon reviewing the Examiner's cited passages, Appellants are unable to determine where Schmeidler teaches storing routing information which can be subsequently accessed by an outbound data stream filter during the processing of an outbound reply. The Examiner's cited passage describes extracting a Universal Resource Name (URN) and performing a URN to URL conversion. Column 14, lines 20-22 further describes that the URN value is forwarded to the CAS server 210. However, the Examiner's cited passages are completely silent as to substantially all of the above-identified limitations.

Reference is made to the following assertion in the seventh bullet on page 7 of the Fifth Office Action:

> accessing, by said particular server, a server-side storage location where information regarding a session between the particular server and the end user device is stored (col.10 lines 45-53).

For ease of reference, the Examiner's cited passage of column 10, lines 45-53 of Schmeidler is reproduced below:

> In addition, the Launcher manages all communications with the CAS, including 1) obtaining from the CAS the address of the RAFT server and the briq path name corresponding to the selected title; 2) obtaining from the CAS a RAFT authorization token and activator necessary to retrieve briq data from the RAFT server and to decrypt the retrieved data; and 3) asking the CAS to refresh the RAFT authorization token and the activator.

As clearly evident, this passage refers to the functions of the Launcher 220, which as already

discussed above, is part of the SCDP client 216 (i.e., the alleged client). As such, Appellants are entirely unclear as to how this passage teaches accessing a server-side storage location by a server. Thus, Schmeidler fails to teach the above-identified additional limitation for which Schmeidler is being relied upon to teach.

Therefore, for the reasons stated above, even if one having ordinary skill in the art were motivated to modify Schmeidler in view of Kunzelman, and Lee the claimed invention would not result.

THE REJECTION OF CLAIMS 6, 10-11, 17, 20-21, AND 27 UNDER 35 U.S.C. § 103 FOR OBVIOUSNESS BASED UPON GUPTA IN VIEW OF SCHMEIDLER AND KUNZELMAN

For convenience of the Honorable Board in addressing the rejections, claims 6, 17, and 22 stand or fall with independent claim 1; and claims 11 and 20-21 stand or fall together with independent claim 10.

With regard to independent claim 10 and the teachings of Gupta, the Examiner asserted the following on page 9 of the Fifth Office Action:

- if a server-side key cookie has been used for storing session information, retrieving a session key from said key cookie (col.12 lines 3-8 and 44-55— retrieving access session cookies);
- if a key cookie was not used for storing session information, retrieving said session key from a control block (col.12 lines 8-18);
- removing all cookies from said response information (col. 13 lines 13-17);
- storing said removed cookies in a predetermined server-side storage area (col.6 lines 28-37, col. 12 lines 48-55, col.13 lines 13-17—cookies are stored and maintained at the server).

The Examiner's cited passage of column 12, lines 3-8, 44-55 is silent with regard to retrieving a session key from a key cookie. The only discussion within these passages are with regard to a

"cookie (or token)" and not as to a session key within the key cookie. Appellants also note that column 13, lines 13-17 does not teach "removing all cookies from said response information," as alleged by the Examiner. This passage cited by the Examiner teaches that the application server caches session information, but this passage is silent as to removing all cookies from the response information. As to the last passage reproduced above, Appellants note that the citation of column 6, lines 28-37 is not only silent as to storing removed cookies, this passage describes what Gupta considers to be prior art. Moreover, although the Examiner's cited passage of column 12, lines 48-55 describes storing a cookie, this passage is silent as to storing a cookie, which has been removed.

Thus, Gupta fails to teach several of the above-identified additional limitation for which Gupta is being relied upon by the Examiner to teach. Therefore, for the reasons stated above, even if one having ordinary skill in the art were motivated to modify Gupta in view of Schmeidler and Kunzelman, the claimed invention would not result.

Conclusion

Based upon the foregoing, Appellants respectfully submit that the Examiner's rejections under 35 U.S.C. §§ 103, 112 are not factually or legally viable. Appellants, therefore, respectfully solicit the Honorable Board to reverse the Examiner's rejections under 35 U.S.C. §§ 103, 112.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in connection with the filing of this paper, including extension of time fees, to Deposit Account 09-

0461, and please credit any excess fees to such deposit account.

Date: March 19, 2007

Respectfully submitted,

/Scott D. Paul/
Scott D. Paul
Registration No. 42,984
Steven M. Greenberg
Registration No. 44,725
CUSTOMER NUMBER 46320

VIII. CLAIMS APPENDIX

1. A method of establishing a persistent relationship between an end user device and a server where the server is one of a plurality of servers managed by a dispatcher and the end user device accesses the server using a uniform resource locator (URL), the method comprising the steps of:

receiving at the dispatcher, a request for information from the end user device;

determining, by the dispatcher, which of the plurality of servers to select for satisfying the request;

creating, at the selected server, a token comprising at least an identifier for the selected server, a date/time stamp, and a key, said key for accessing a server-side storage area for information regarding the persistent relationship and the end user device;

inserting the token into the URL; and,

sending, by the selected server to the client device, a response with the token inserted into the URL.

2. A method as claimed in claim 1 wherein said token is encoded using a modified Base64 encoding.

3. A method as claimed in claim 1 wherein said token has a checksum or hash verification field.

4. A method as claimed in claim 3 wherein said hash is a SHA-1 hash computed over

said identifier for said selected server, said date/time stamp, and said key.

5. A method as claimed in claim 3 wherein said checksum or hash is encoded using a modified Base64 encoding.

6. A method as claimed in claim 1 wherein said information regarding said persistent relationship is stored as a cookie on said server.

7. A method of routing a request by an end user device to a particular one of a plurality of redundant servers residing behind a network dispatching mechanism, said methods comprising the steps of:

receiving, at the network dispatching mechanism, a request for information indicated by a uniform resource locator (URL);

determining, at the network dispatching mechanism, if said URL contains a valid routing token;

if said URL contains a valid routing token, further determining, at the network dispatching mechanism, if a session binding indicated by said routing token is old;

if said URL contains a valid routing token and said routing token is not old, forwarding, by said network dispatching mechanism, the request, including the URL, to the particular server indicated by said valid routing token;

removing, by said particular server, said valid routing information from the URL;

storing, by said particular server, said routing information removed from said valid routing token, where said valid routing information can be accessed subsequently by an

outbound data stream filter during the processing of an outbound reply related to said request;

accessing, by said particular server, a server-side storage location where session information regarding a session between the particular server and the end user device is stored; and,

inserting, by said particular server, said accessed session information into said request.

8. A method as claimed in claim 7 wherein additional filtering of the URL is done prior to the forwarding step.

9. A method as claimed in claim 1 wherein all filtering is performed within the dispatcher.

10. A method of sending information to a requesting end user from an application over a session wherein said application resides at one of a plurality of redundant servers residing behind a network dispatcher, said method comprising the steps of:

receiving response information from said application, said response information including a URL (uniform resource locator);

determining if a server-side key cookie has been used for storing session information between said end user and said application;

if a server-side key cookie has been used for storing session information, retrieving a session key from said key cookie;

if a key cookie was not used for storing session information, retrieving said session key from a control block;

removing all cookies from said response information;

storing said removed cookies in a predetermined server-side storage area;

updating said URL to indicate the removal of said cookies;

creating a sticky routing string;

updating a date/time stamp in said sticky routing string;

inserting said sticky routing string into said URL; and,

transmitting said response information, including said URL to said end user.

11. A method as claimed in claim 10 wherein, prior to said determining step, said response information is transmitted from said application through one or more filters.

12. A computer program product having computer readable code means of establishing a persistent relationship between an end user device and a server where the server is one of a plurality of servers managed by a dispatcher and the end user device accesses the server using a uniform resource locator (URL), the computer program product comprising:

computer readable code means of receiving at the dispatcher, a request for information from the end user device;

computer readable code means of determining by the dispatcher, which of the plurality of servers to select for satisfying the request;

computer readable code means of creating, at the selected server, a token comprising at least an identifier for the selected server, a data/time stamp, and a key, said key for accessing a server-side storage area for information regarding the persistent relationship and the end user device;

computer readable code means of inserting the token into the URL; and,

computer readable code means of sending, by the selected server to the client device, a response with the token inserted into the URL.

13. A computer program product as claimed in claim 12 wherein said token is encoded using a modified Base64 encoding.

14. A computer program product as claimed in claim 12 wherein said token has a checksum or hash verification field.

15. A computer program product as claimed in claim 14 wherein said hash is a SHA-1 hash computed over said identifier for said selected server, said date/time stamp, and said key.

16. A computer program product as claimed in claim 14 wherein said checksum or hash is encoded using a modified Base64 encoding.

17. A computer program product as claimed in claim 12 wherein said information regarding said persistent relationship is stored as a cookie on said server.

18. A computer program product having computer readable code means for routing a request by an end user device to a particular one of a plurality of redundant servers residing behind a network dispatching mechanism, said computer program product comprising:

computer readable program code for receiving, at the network dispatching mechanism, a

request for information indicated by a uniform resource locator (URL);

computer readable program code for determining, at the network dispatching mechanism, if said URL contains a valid routing token;

if said URL contains a valid routing token, computer readable program code for further determining, at the network dispatching mechanism, if a session binding indicated by said routing token is old;

if said URL contains a valid routing token and said routing token is not old, computer readable program code for forwarding, by said network dispatching mechanism, the request, including the URL, to the particular server indicated by said valid routing token;

computer readable program code for removing, by said particular server, said valid routing information from the URL;

computer readable program code for storing, by said particular server, said routing information removed from said valid routing token, where said valid routing information can be accessed subsequently by an outbound data stream filter during the processing of an outbound reply related to said request;

computer readable program code for accessing, by said particular server, a server-side storage location where session information regarding a session between the particular server and the end user device is stored; and,

computer readable program code for inserting, by said particular server, said accessed session information into said request.

19. The computer program product as claimed in claim 18 wherein additional filtering of the URL is done prior to the forwarding step.

20. A computer program product having computer readable code means of sending information to a requesting end user from an application over a session wherein said application resides at one of a plurality of redundant servers residing behind a network dispatcher, said computer program product comprising:

computer readable programming means of receiving response information from said application, said response information including a URL (uniform resource locator);

computer readable programming means of determining if a server-side key cookie has been used for storing session information between said end user and said application;

if a server-side key cookie has been used for storing session information, computer readable programming means of retrieving a session key from said key cookie;

if a key cookie was not used for storing session information, computer readable programming means of retrieving said session key from a control block;

computer readable programming means of removing all cookies from said response information;

computer readable programming means of storing said removed cookies in a predetermined server-side storage area;

computer readable programming means of updating said URL to indicate the removal of said cookies;

computer readable programming means of creating a sticky routing string;

computer readable programming means of updating a date/time stamp in said sticky routing string;

computer readable programming means of inserting said sticky routing string into said

URL; and,

computer readable programming means of transmitting said response information, including said URL to said end user.

21. A computer program product as claimed in claim 20 wherein, prior to said determining step, said response information is transmitted from said application through one or more filters.

22. A network dispatcher for establishing a persistent relationship between an end user device and a server where the server is one of a plurality of servers managed by said network dispatcher comprising:

means for receiving a request for information from said end user device, said request for information including a uniform resource locator (URL);

means for determining which of the plurality of servers to select for satisfying said request for information;

means for creating, at said selected server, a token comprising at least an identifier for the selected server, a date/time stamp, and a key, said key for accessing a server-side storage area for information regarding the persistent relationship and the end user device;

means for inserting the token into the URL; and,

means for sending, by the selected server, a response with the token inserted into the URL to the client device.

23. A network dispatcher as claimed in claim 22 wherein said token is encoded using a

modified Base64 encoding.

24. A network dispatcher as claimed in claim 22 wherein said token has a checksum or hash verification field.

25. A network dispatcher as claimed in claim 24 wherein said hash is a SHA-1 has computed over said identifier for said selected server, said date/time stamp, and said key.

26. A network dispatcher as claimed in claim 24 wherein said checksum or hash is encoded using a modified Base64 encoding.

27. A network dispatcher as claimed in claim 22 wherein said information regarding the persistent relationship is stored as a cookie on said server.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellants in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellants are unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.